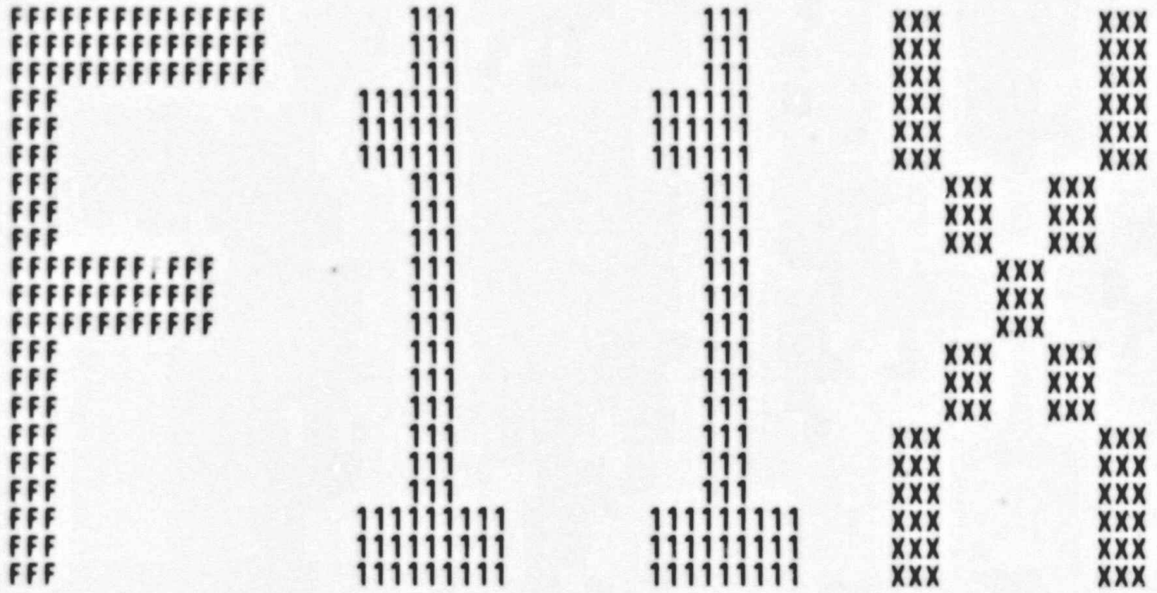

_$2

Symb

IOCS
IO_C
IO_C
IO_D
IO_F
IO_S
KICL

KILL
KILL
LB_B
LB_D
LB_F
LB_K
LB_L
LOCA
LOCA
LOCK

LOCK
LOCK
LOCK
LOC_
LOC_
L_CO
L_CO
L_DA
L_DA
MAIN
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE

MAKE
MAKE
MAP_
MAP_

MAP
MARK
MARK
MARK
MARK
MARK

FILEID**SNDBAD

```
  SSSSSSSS  NN      NN  DDDDDDDD    BBBBBBBB      AAAAAA    DDDDDDDD
  SSSSSSSS  NN      NN  DDDDDDDD    BBBBBBBB      AAAAAA    DDDDDDDD
SS          NN      NN  DD      DD  BB      BB   AA      AA  DD      DD
SS          NN      NN  DD      DD  BB      BB   AA      AA  DD      DD
SS          NNNN    NN  DD      DD  BB      BB   AA      AA  DD      DD
SS          NNNN    NN  DD      DD  BB      BB   AA      AA  DD      DD
  SSSSSS    NN  NN  NN  DD      DD  BBBBBBBB   AA      AA  DD      DD
  SSSSSS    NN  NN  NN  DD      DD  BBBBBBBB   AA      AA  DD      DD
        SS  NN    NNNN  DD      DD  BB      BB   AAAAAAAAAA  DD      DD
        SS  NN    NNNN  DD      DD  BB      BB   AAAAAAAAAA  DD      DD
        SS  NN      NN  DD      DD  BB      BB   AA      AA  DD      DD
        SS  NN      NN  DD      DD  BB      BB   AA      AA  DD      DD
SSSSSSSS    NN      NN  DDDDDDDD    BBBBBBBB     AA      AA  DDDDDDDD
SSSSSSSS    NN      NN  DDDDDDDD    BBBBBBBB     AA      AA  DDDDDDDD

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

```
;     1            0001  0 MODULE SNDBAD (
;     2            0002  0                 LANGUAGE (BLISS32),
;     3            0003  0                 IDENT = 'V04-000'
;     4            0004  0                 ) =
;     5            0005  1 BEGIN
;     6            0006  1
;     7            0007  1 !
;     8            0008  1 !*****************************************************************************
;     9            0009  1 !*                                                                           *
;    10            0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    11            0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    12            0012  1 !*  ALL RIGHTS RESERVED.                                                      *
;    13            0013  1 !*                                                                           *
;    14            0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    15            0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;    16            0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    17            0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    18            0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    19            0019  1 !*  TRANSFERRED.                                                              *
;    20            0020  1 !*                                                                           *
;    21            0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    22            0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    23            0023  1 !*  CORPORATION.                                                              *
;    24            0024  1 !*                                                                           *
;    25            0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    26            0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    27            0027  1 !*                                                                           *
;    28            0028  1 !*                                                                           *
;    29            0029  1 !*****************************************************************************
;    30            0030  1
;    31            0031  1 !++
;    32            0032  1 !
;    33            0033  1 ! FACILITY:  F11ACP Structure Level 2
;    34            0034  1 !
;    35            0035  1 ! ABSTRACT:
;    36            0036  1 !
;    37            0037  1 !       This routine sends a message to the bad block analysis program to
;    38            0038  1 !       deal with a file that is marked bad.
;    39            0039  1 !
;    40            0040  1 ! ENVIRONMENT:
;    41            0041  1 !
;    42            0042  1 !       STARLET operating system, including privileged system services
;    43            0043  1 !       and internal exec routines.
;    44            0044  1 !
;    45            0045  1 !--
;    46            0046  1 !
;    47            0047  1 !
;    48            0048  1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  26-May-1978  14:50
;    49            0049  1 !
;    50            0050  1 ! MODIFIED BY:
;    51            0051  1 !
;    52            0052  1 !       V03-007 CDS0006         Christian D. Saether     2-July-1984
;    53            0053  1 !               Need to have enhanced privileges for $ASSIGN also.
;    54            0054  1 !
;    55            0055  1 !       V03-006 CDS0005         Christian D. Saether    20-Jun-1984
;    56            0056  1 !               Raise/lower process biocnt and astcnt around
;    57            0057  1 !               $qio to mailbox so that it does not fail for
```

```
;    58          0058  1 !                   lack of those quotas.
;    59          0059  1 !
;    60          0060  1 !       V03-005 CDS0004         Christian D. Saether    30-Dec-1983
;    61          0061  1 !               Use L_NORM linkage and BIND_COMMON macro.
;    62          0062  1 !
;    63          0063  1 !       V03-004 CDS0003         Christian D. Saether     5-Oct-1983
;    64          0064  1 !               Fix bug restoring privs to PCB.
;    65          0065  1 !
;    66          0066  1 !       V03-003 CDS0002         Christian D. Saether    13-Jan-1983
;    67          0067  1 !               Separately save and restore PHD privs.
;    68          0068  1 !
;    69          0069  1 !       V03-002 CDS0001         Christian D. Saether    28-Dec-1982
;    70          0070  1 !               Give the process DETACH and SETPRV for the CREPRC of
;    71          0071  1 !               the bad block scanner, and BYPASS to assign channel
;    72          0072  1 !               to bad block scanner mailbox.
;    73          0073  1 !               Also use PIC_DESC instead of DESCRIPTOR.
;    74          0074  1 !
;    75          0075  1 !       V03-001 LMP0037         L. Mark Pilant,         28-Jun-1982  15:10
;    76          0076  1 !               Remove the addressing mode module switch.
;    77          0077  1 !
;    78          0078  1 !       V02-003 LMP0013         L. Mark Pilant,         15-Mar-1981 16:20
;    79          0079  1 !               Remove unused and unneeded sorage (to fix Linker truncation
;    80          0080  1 !               errors).
;    81          0081  1 !
;    82          0082  1 !       V02-002 ACG0230         Andrew C. Goldstein,    24-Dec-1981  0:17
;    83          0083  1 !               Go to general mode addressing for externals
;    84          0084  1 !
;    85          0085  1 !       V02-001 ACG0167         Andrew C. Goldstein,    16-Apr-1980  19:28
;    86          0086  1 !               Previous revision history moved to F11B.REV
;    87          0087  1 !**
;    88          0088  1
;    89          0089  1
;    90          0090  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    91          0091  1 REQUIRE 'SRC$:FCPDEF.B32';
```

```
;    93          1082  1 GLOBAL ROUTINE SEND_BADSCAN (FID) : L_NORM NOVALUE =
;    94          1083  1
;    95          1084  1 !++
;    96          1085  1 !
;    97          1086  1 ! FUNCTIONAL DESCRIPTION:
;    98          1087  1 !
;    99          1088  1 !       This routine sends a message to the bad block analysis program to
;   100          1089  1 !       deal with a file that is marked bad.
;   101          1090  1 !
;   102          1091  1 !
;   103          1092  1 ! CALLING SEQUENCE:
;   104          1093  1 !       SEND_BADSCAN (ARG1)
;   105          1094  1 !
;   106          1095  1 ! INPUT PARAMETERS:
;   107          1096  1 !       ARG1: address of file ID of file
;   108          1097  1 !
;   109          1098  1 ! IMPLICIT INPUTS:
;   110          1099  1 !       CURRENT_UCB: UCB of device containing file
;   111          1100  1 !
;   112          1101  1 ! OUTPUT PARAMETERS:
;   113          1102  1 !       NONE
;   114          1103  1 !
;   115          1104  1 ! IMPLICIT OUTPUTS:
;   116          1105  1 !       NONE
;   117          1106  1 !
;   118          1107  1 ! ROUTINE VALUE:
;   119          1108  1 !       NONE
;   120          1109  1 !
;   121          1110  1 ! SIDE EFFECTS:
;   122          1111  1 !       bad block scan process started
;   123          1112  1 !
;   124          1113  1 !--
;   125          1114  1
;   126          1115  2 BEGIN
;   127          1116  2
;   128          1117  2 MAP
;   129          1118  2         FID             : REF BBLOCK;   ! file ID argument
;   130          1119  2
;   131          1120  2 LOCAL
;   132          1121  2         PTR             : REF BBLOCK,
;   133          1122  2         SAVE_PRIV       : VECTOR [4],
;   134          1123  2         DESC0           : VECTOR [2],   ! descriptor
;   135          1124  2         MBX_CHANNEL     : WORD;         ! channel number for mailbox
;   136          1125  2
;   137          1126  2 BIND_COMMON;
;   138          1127  2
;   139          1128  2 EXTERNAL
;   140          1129  2         CTL$GL_PCB      : ADDRESSING_MODE (GENERAL),
;   141          1130  2         CTL$GL_PHD      : ADDRESSING_MODE (GENERAL);
;   142          1131  2
;   143          1132  2 ! Assign a channel to the bad block scanner mailbox. Note that we simply
;   144          1133  2 ! give up on errors - the file will be left marked for delete and bad and
;   145          1134  2 ! can be picked up and retried later.
;   146          1135  2 !
;   147          1136  2
;   148          1137  2 PIC_DESC ('ACP$BADBLOCK_MBX', DESC0);
;   149          1138  2
```

```
;  150          1139  2 ! We don't need to raise bio, ast counts for the assign, only need detach
;  151          1140  2 ! for the creprc, but just do it all once for all the services that follow.
;  152          1141  2 !
;  153          1142  2
;  154          1143  2 PTR = .CTL$GL_PCB;
;  155          1144  2 PTR [PCB$W_BIOCNT] = .PTR [PCB$W_BIOCNT] + 1;
;  156          1145  2 PTR [PCB$W_ASTCNT] = .PTR [PCB$W_ASTCNT] + 1;
;  157          1146  2 SAVE_PRIV [0] = .(PTR [PCB$Q_PRIV]);
;  158          1147  2 SAVE_PRIV [1] = .(PTR [PCB$Q_PRIV]+4);
;  159          1148  2 BBLOCK [ PTR [PCB$Q_PRIV], PRV$V_DETACH] = 1;
;  160          1149  2 BBLOCK [ PTR [PCB$Q_PRIV], PRV$V_SETPRV] = 1;
;  161          1150  2 BBLOCK [ PTR [PCB$Q_PRIV], PRV$V_BYPASS] = 1;
;  162          1151  2
;  163          1152  2 PTR = .CTL$GL_PHD;
;  164          1153  2 SAVE_PRIV [2] = .(PTR [PHD$Q_PRIVMSK]);
;  165          1154  2 SAVE_PRIV [3] = .(PTR [PHD$Q_PRIVMSK]+4);
;  166          1155  2 BBLOCK [ PTR [PHD$Q_PRIVMSK], PRV$V_DETACH] = 1;
;  167          1156  2 BBLOCK [ PTR [PHD$Q_PRIVMSK], PRV$V_SETPRV] = 1;
;  168          1157  2 BBLOCK [ PTR [PHD$Q_PRIVMSK], PRV$V_BYPASS] = 1;
;  169          1158  2
;  170        P 1159  2 IF NOT $ASSIGN (CHAN = MBX_CHANNEL,
;  171          1160  3                 DEVNAM = DESC0)
;  172          1161  2 THEN
;  173          1162  3     BEGIN
;  174          1163  3     (PTR [PHD$Q_PRIVMSK]) = .SAVE_PRIV [2];
;  175          1164  3     (PTR [PHD$Q_PRIVMSK]+4) = .SAVE_PRIV [3];
;  176          1165  3
;  177          1166  3     PTR = .CTL$GL_PCB;
;  178          1167  3     PTR [PCB$W_BIOCNT] = .PTR [PCB$W_BIOCNT] - 1;
;  179          1168  3     PTR [PCB$W_ASTCNT] = .PTR [PCB$W_ASTCNT] - 1;
;  180          1169  3     (PTR [PCB$Q_PRIV]) = .SAVE_PRIV [0];
;  181          1170  3     (PTR [PCB$Q_PRIV]+4) = .SAVE_PRIV [1];
;  182          1171  3
;  183          1172  3     RETURN
;  184          1173  2     END;
;  185          1174  2
;  186          1175  2 ! Send the message. Then attempt to create the bad block scan process. If one
;  187          1176  2 ! is already active, the create will fail due to duplicate process names,
;  188          1177  2 ! and the message will simply be queued.
;  189          1178  2 !
;  190          1179  2
;  191        P 1180  2 IF $QIO (CHAN = .MBX_CHANNEL,
;  192        P 1181  2          FUNC = IO$_WRITELBLK OR IO$M_NOW,
;  193        P 1182  2          EFN  = MBX_EFN,
;  194        P 1183  2          P1   =
;  195        P 1184  2                 BEGIN
;  196        P 1185  2                 ! Construct the message in the message buffer.
;  197        P 1186  2                 !
;  198        P 1187  2
;  199        P 1188  2                 LOCAL
;  200        P 1189  2                     MESSAGE     : BBLOCK [BBS$C_LENGTH]; ! message buffer
;  201        P 1190  2
;  202        P 1191  2                 CH$FILL (0, BBS$C_LENGTH, MESSAGE);
;  203        P 1192  2                 MESSAGE[BBS$B_MSGTYPE] = MSG$_SCANBAD;
;  204        P 1193  2                 MESSAGE[BBS$W_SEQUENCE] = 0;
;  205        P 1194  2                 MESSAGE[BBS$L_UCB] = .CURRENT_UCB;
;  206        P 1195  2                 CH$MOVE (FID$C_LENGTH, .FID, MESSAGE[BBS$W_FID]);
```

```
;   207          P 1196  2                    MESSAGE
;   208          P 1197  2                    END,
;   209          P 1198  2                P2    = BBS$C_LENGTH
;   210            1199  3                )
;   211            1200  3
;   212            1201  2 THEN
;   213            1202  3     BEGIN
;   214            1203  3
;   215            1204  3     LOCAL
;   216            1205  3         DESC1           : VECTOR [2],   ! descriptor
;   217            1206  3         DESC2           : VECTOR [2];   ! descriptor
;   218            1207  3
;   219            1208  3     PIC_DESC ('SYS$SYSTEM:BADBLOCK.EXE', DESC0);
;   220            1209  3     PIC_DESC (' TTA1:', DESC1);
;   221            1210  3     PIC_DESC ('BADBLOCK_SCAN', DESC2);
;   222            1211  3
;   223          P 1212  3     $CREPRC (
;   224          P 1213  3         IMAGE  = DESC0,
;   225          P 1214  3         INPUT  = DESC1,
;   226          P 1215  3         OUTPUT = DESC1,
;   227          P 1216  3         ERROR  = DESC1,
;   228          P 1217  3         PRVADR = UPLIT (-1, -1),
;   229          P 1218  3         PRCNAM = DESC2,
;   230          P 1219  3         BASPRI = 4,
;   231          P 1220  3         UIC    = 1^16 + 3
;   232            1221  3         );
;   233            1222  3
;   234            1223  2     END;
;   235            1224  2
;   236            1225  2 $DASSGN (CHAN = .MBX_CHANNEL);
;   237            1226  2
;   238            1227  2 (PTR [PHD$Q_PRIVMSK]) = .SAVE_PRIV [2];
;   239            1228  2 (PTR [PHD$Q_PRIVMSK]+4) = .SAVE_PRIV [3];
;   240            1229  2
;   241            1230  2 PTR = .CTL$GL_PCB;
;   242            1231  2 PTR [PCB$W_BIOCNT] = .PTR [PCB$W_BIOCNT] - 1;
;   243            1232  2 PTR [PCB$W_ASTCNT] = .PTR [PCB$W_ASTCNT] - 1;
;   244            1233  2 (PTR [PCB$Q_PRIV]) = .SAVE_PRIV [0];
;   245            1234  2 (PTR [PCB$Q_PRIV]+4) = .SAVE_PRIV [1];
;   246            1235  2
;   247            1236  1 END;                                    ! end of routine SEND_BADSCAN


                                                        .TITLE  SNDBAD
                                                        .IDENT  \V04-000\

                                                        .PSECT  $CODE$,NOWRT,2

42 4D 5F 4B 43 4F 4C 42 44 41 42 24 50 43 41  00000 P.AAA:  .ASCII  \ACP$BADBLOCK_MBX\                    ;
                                              58  0000F                                                  ;
42 44 41 42 3A 4D 45 54 53 59 53 24 53 59 53  00010 P.AAB:  .ASCII  \SYS$SYSTEM:BADBLOCK.EXE\<0>         ;
                  00 45 58 45 2E 4B 43 4F 4C  0001F                                                     ;
                     00 00 3A 31 41 54 54 5F  00028 P.AAC:  .ASCII  \ TTA1:\<0><0>                       ;
00 00 4E 41 43 53 5F 4B 43 4F 4C 42 44 41 42  00030 P.AAD:  .ASCII  \BADBLOCK_SCAN\<0><0><0>            ;
                                              00  0003F                                                 ;
                                 FFFFFFFF FFFFFFFF  00040 P.AAE:  .LONG   -1, -1                        ;
```

```
                                                         .EXTRN  CTL$GL_PCB, CTL$GL_PHD
                                                         .EXTRN  SYS$ASSIGN, SYS$QIO
                                                         .EXTRN  SYS$CREPRC, SYS$DASSGN

                                     01FC 00000          .ENTRY  SEND_BADSCAN, Save R2,R3,R4,R5,R6,R7,R8      ; 1082
                 58 00000000G    00   9E 00002           MOVAB   CTL$GL_PCB, R8                               ;
                 57          AC  AF   9E 00009           MOVAB   P.AAA, R7                                    ;
                 5E              30   C2 0000D           SUBL2   #48, SP                                      ;
          18     AE              10   D0 00010           MOVL    #16, DESC0                                   ; 1137
          1C     AE              67   9E 00014           MOVAB   P.AAA, DESC0+4                               ;
                 56              68   D0 00018           MOVL    CTL$GL_PCB, PTR                              ; 1143
                             3A  A6   B6 0001B           INCW    58(PTR)                                      ; 1144
                             38  A6   B6 0001E           INCW    56(PTR)                                      ; 1145
          20     AE        0084  C6   7D 00021           MOVQ    132(PTR), SAVE_PRIV                          ; 1146
        0084     C6 20004020     8F   C8 00027           BISL2   #536887328, 132(PTR)                         ; 1150
                 56 00000000G    00   D0 00030           MOVL    CTL$GL_PHD, PTR                              ; 1152
          28     AE              66   7D 00037           MOVQ    (PTR), SAVE_PRIV+8                           ; 1153
                 66 20004020     8F   C8 0003B           BISL2   #536887328, (PTR)                            ; 1157
                                 7E   7C 00042           CLRQ    -(SP)                                        ; 1160
                             08  AE   9F 00044           PUSHAB  MBX_CHANNEL                                  ;
                             24  AE   9F 00047           PUSHAB  DESC0                                        ;
   00000000G     00              04   FB 0004A           CALLS   #4, SYS$ASSIGN                               ;
                 03              50   E8 00051           BLBS    R0, 1$                                       ;
                               0088   31 00054           BRW     3$                                           ;
                                 7E   7C 00057    1$:    CLRQ    -(SP)                                        ; 1199
                                 7E   7C 00059           CLRQ    -(SP)                                        ;
                                 12   DD 0005B           PUSHL   #18                                          ;
12        00     6E              00   2C 0005D           MOVC5   #0, (SP), #0, #18, MESSAGE                   ;
                             18  AE      00062                                                                ;
          18     AE              28   90 00064           MOVB    #40, MESSAGE                                 ;
                             1C  AE   B4 00068           CLRW    MESSAGE+4                                    ;
          20     AE          94  AA   D0 0006B           MOVL    -108(BASE), MESSAGE+8                        ;
24   AE   04     BC              06   28 00070           MOVC3   #6, @FID, MESSAGE+12                         ;
                             18  AE   9F 00076           PUSHAB  MESSAGE                                      ;
                                 7E   7C 00079           CLRQ    -(SP)                                        ;
                                 7E   D4 0007B           CLRL    -(SP)                                        ;
          7E                 60  8F   9A 0007D           MOVZBL  #96, -(SP)                                   ;
          7E                 28  AE   3C 00081           MOVZWL  MBX_CHANNEL, -(SP)                           ;
                                 1E   DD 00085           PUSHL   #30                                          ;
   00000000G     00              0C   FB 00087           CALLS   #12, SYS$QIO                                 ;
                 44              50   E9 0008E           BLBC    R0, 2$                                       ;
          18     AE              17   D0 00091           MOVL    #23, DESC0                                   ; 1208
          1C     AE          10  A7   9E 00095           MOVAB   P.AAB, DESC0+4                               ;
          10     AE              06   D0 0009A           MOVL    #6, DESC1                                    ; 1209
          14     AE          28  A7   9E 0009E           MOVAB   P.AAC, DESC1+4                               ;
          08     AE              0D   D0 000A3           MOVL    #13, DESC2                                   ; 1210
          0C     AE          30  A7   9E 000A7           MOVAB   P.AAD, DESC2+4                               ;
                                 7E   7C 000AC           CLRQ    -(SP)                                        ; 1221
                                 7E   D4 000AE           CLRL    -(SP)                                        ;
                    00010003     8F   DD 000B0           PUSHL   #65539                                       ;
                                 04   DD 000B6           PUSHL   #4                                           ;
                             1C  AE   9F 000B8           PUSHAB  DESC2                                        ;
                                 7E   D4 000BB           CLRL    -(SP)                                        ;
                             40  A7   9F 000BD           PUSHAB  P.AAE                                        ;
                             30  AE   9F 000C0           PUSHAB  DESC1                                        ;
                             34  AE   9F 000C3           PUSHAB  DESC1                                        ;
                             38  AE   9F 000C6           PUSHAB  DESC1                                        ;
```

```
                              44    AE   9F 000C9       PUSHAB  DESCO                          ;
                                         7E   D4 000CC       CLRL    -(SP)                          ;
              00000000G  00              0D   FB 000CE       CALLS   #13, SYS$CREPRC                ;
                         7E              6E   3C 000D5 2$:   MOVZWL  MBX_CHANNEL, -(SP)             ; 1225
              00000000G  00              01   FB 000D8       CALLS   #1, SYS$DASSGN                 ;
                         66       28    AE   7D 000DF 3$:   MOVQ    SAVE_PRIV+8, (PTR)             ; 1227
                         56              68   D0 000E3       MOVL    CTL$GL_PCB, PTR                ; 1230
                                  3A    A6   B7 000E6       DECW    58(PTR)                        ; 1231
                                  38    A6   B7 000E9       DECW    56(PTR)                        ; 1232
                   0084  C6       20    AE   7D 000EC       MOVQ    SAVE_PRIV, 132(PTR)            ; 1233
                                         04 000F2       RET                                    ; 1236

; Routine Size:  243 bytes,    Routine Base:  $CODE$ + 0048


;  248         1237  1
;  249         1238  1 END
;  250         1239  0 ELUDOM




;                                   PSECT SUMMARY
;
;       Name                        Bytes                         Attributes
;
;   $CODE$                              315  NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)




;                              Library Statistics
;
;                                          -------- Symbols --------      Pages        Processing
;       File                               Total    Loaded   Percent      Mapped       Time
;
;   _$255$DUA28:[SYSLIB]LIB.L32;1          18619       38         0        1000          00:01.9




;                                    COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:SNDBAD/OBJ=OBJ$:SNDBAD MSRC$:SNDBAD/UPDATE=(ENH$:SNDBAD)

; Size:            243 code + 72 data bytes
; Run Time:           00:20.2
; Elapsed Time:       00:38.3
; Lines/CPU Min:       3674
; Lexemes/CPU-Min: 51081
; Memory Used:   248 pages
; Compilation Complete
```

SCHFCB
LIS

SNDSMB
LIS

SHFDIR
LIS

SNDERR
LIS

TRUNC
LIS

FAL

FAL
MAP

SELVOL
LIS

DAPDEF
MDL

SMALOC
LIS

SNDBAD
LIS

SWITVL
LIS

WITURN
LIS